Remarks/Arguments

The Applicants respectfully request further examination and reconsideration in view of the amendments made above and the arguments set forth below. Claims 1-45, 47-52, and 59-71 were pending. Claims 46 and 53-58 were previously canceled. Within the Previous Office Action, mailed March 16, 2009, Claims 1-45, 47-52, and 59-71 have been rejected under 35 U.S.C. § 103(a). By way of the above amendments, Claims 1, 16, 26, 36, 48, and 70 have been amended, and Claims 72 and 73 have been added. Accordingly, Claims 1-45, 47-52 and 59-73 are now pending.

Embodiments

In accordance with embodiments of the presently claimed invention, a virtual memory facility is modified so that all incoming data is decrypted and all outgoing data is encrypted: a vnode in accordance with the embodiments is modified to encrypt and decrypt data entering and leaving kernel space. The vnode layer contains drivers used to encrypt and decrypt *directly*, such as by using vnode operations. The vnode layer does not encrypt and decrypt indirectly, such as by making calls to another, separate layer.

Directly encrypting and decrypting in this way is taught throughout the Specification. As examples: page 61, lines 23-25, and Figure 28 of the Present Specification together explain that encryption occurs in the vnode, between high-level and low-level vnode operations, "as the seg_map driver copies data from user memory to kernel memory"; Figure 29 shows vnode operations directly calling or being called by encryption and decryption functions; page 45, lines 17-18, state, "encryption drivers are *integrated into* the vnode interface structure" of UNIX source code (italics added); page 59, line 25, to page 60, line 1, explains how a vnode layer *itself* is configured to encrypt and decrypt data.

In accordance with the embodiments, encryption and decryption can be performed faster than prior art systems that must call a separate layer to perform these functions.

Riedel

U.S. Patent No. 7,313,694 to Riedel ("Riedel"), cited within the Previous Office Action, is directed to a technique for secure file access control via directory encryption. Riedel discloses encrypting filenames to protect them in the event a server is untrustworthy, such as in a distributed computing environment. Riedel also discloses encrypting filenames in a directory structure without otherwise changing the directory structure. (Riedel, Abstract)

Zadok

Zadok, "Cryptfs: A Stackable Vnode Level Encryption File System" ("Zadok"), cited within the Previous Office Action, discloses a "stackable" vnode interface that communicates with a separate Cryptfs layer, used to encrypt and decrypt data. As Zadok shows in its Figure 1 and explains in the accompanying text, "[S]ystem calls are translated into vnode level calls, and those invoke their Cryptfs equivalents. Cryptfs again invokes generic vnode operations, and the latter call their respective 'lower level' file system specific operations such as UFS." (Zadok, page 2, col. 1, 3d full paragraph). This specific structure–a separate Cryptfs layer–is critical for Zadok's system to be portable, a stated design goal. (Zadok, page 2, col. 2, 5th full paragraph)

Response to arguments in the June 1, 2009, Office Action.

A. Arguments directed to the vnode layer.

Within the Office Action mailed June 1, 2009 (the June Office Action), it is stated, "It is also noted that there is no definition provided for the term integrated that would differentiate over the interfacing of the prior art *nor is there any claim language that uses the term integrated* or any of its derivatives" (italics added). The Applicants respectfully disagree.

Before this amendment, Claim 36 recited, in part, "the kernel code comprising a virtual node *integrated with drivers* to encrypt the clear data file"; Claim 48 recited, in part, "the kernel comprising a virtual node *integrated with drivers* to decrypt the first and second elements"; Claim 70 recited, in part, "a kernel comprising a virtual node *integrated with drivers* to encrypt and decrypt data"; and Claim 71 recited, in part, "the kernel code comprises a virtual node *integrated with drivers* to use a symmetric key to encrypt the clear data" (italics added to all claims).

To further prosecution, Claim 36 has been amended to further recite, among other things, a vnode that comprises drivers to *directly* encrypt clear data. Similarly, Claim 48 has been amended to further recite that the vnode layer *directly* encrypts and decrypts. This added limitation clearly differentiates the claims over the cited prior art.

Within the June Office Action it is also stated (italics added):

> Zadok uses stacking and interface as complimentary terms. Stacking provides for more than one vnode interface. The vnode does the encryption and decryption through system calls that are translated into vnode level calls, *and invoke the interface cryptfs modules*. The cryptfs module adds functionality to the vnode; it does not operate separately from the vnode.

The June Office Action thus argues that a vnode and the cryptfs module operate together, not separately. However the two operate, it is clear that Zadok's vnode layer calls Cryptfs to encrypt and decrypt; Zadok's vnode does not encrypt or decrypt *directly*, such as by calling drivers *within the vnode* itself. The vnode layer in Zadok does not comprise drivers to encrypt and decrypt.

B. Arguments directed to generating encryption keys.

Within the June Office Action, it is stated (italics added):

> The Examiner concedes that the Blaze reference *does not have the same verbiage* as the claim limitations. However, the Examiner sees the functionality of the claim language in the Blaze reference. A pass key and a data file name are combined, *in some way*, to generate another key. The data file name is seen as a random number and the procedure merely randomizes the pass key. Similarly, Blaze takes keys that are turned into arbitrary-length "passphrases" (random number) that are used to generate the cryptographic keys.

Here, the Examiner generalizes what the claims recite, while ignoring their specific language. For example, Claim 16 recites a key engine that:

- receives a pass key and a data file name to generate an encrypted data file name key;
- uses the encrypted data file name key and data file contents to generate an encrypted data file contents key;
- encrypts the data file contents with an encrypting data file contents key to generate encrypted data file contents; and
- encrypts the data file name with an encrypting data file name key to generate an encrypted data file name.

In comparison, Blaze (Blaze, "A Cryptographic File System for Unix"), cited within the Previous Office Action, discloses the following steps, where X–>Y refers to using X to generate Y, and [] refers to an encryption operation:

Passphrase–> Key1 + Key2
Key1–>RandomBitMask
For each Block of Data: RandomBitMask XOR Block–>masked data
For each Block of Data: Key2[masked data]–>encrypted data

(See Blaze, page 13, col. 1, last paragraph, to col. 2., first paragraph)

Blaze discloses using a pass phrase to generate two keys to encrypt data. One of these keys is used to form a pseudo-random bitmask, which is applied to blocks of data. Blaze does not disclose using a pass key and a *data file name* to generate a file name key, in accordance with the claimed invention. In accordance with the claims, the name of the file itself is used to encrypt and, ultimately, decrypt the name.

Above, the examiner argues that a data file name functions just like a random number. Even if a file name were the functional equivalent of a random number (it is not) the random number is not used to encrypt itself. Thus, Blaze does not disclose an element of the claims.

As they were required to do (35 U.S.C. § 112, ¶ 2), the Applicants particularly defined embodiments of their invention by reciting a "file name," not a pseudo-random number. A file name is different from a random number. The differences cannot be ignored when determining patentability: "All words in a claim must be considered in judging the patentability of that claim against the prior art." M.P.E.P. § 2143.03 (quoting *In re Wilson*, 424 F.2d 1382, 1385 (C.C.P.A. 1970)).

Specifically, a file name is assigned to a file, usually by a user. A file name can be selected to help a user remember the contents of a file or to show the relationships between different files *(e.g.*, doc1.header, doc1.exe, changes.bwt.doc2doc1); a pseudo-random number cannot be used this way. A file name is used to locate files, such as by traversing a directory, such as to open, edit, delete, and move the files; a pseudo-random number is not used in this way. When used to generate a key, a file name is a known "seed," readily determined for encryption, unique for each file in a directory; a pseudo-random number must be generated by a time-consuming algorithm and checked to ensure its uniqueness. In short, a file name is different from a random number.

Blaze does not disclose a second element recited in the claims. For example, Claim 16 recites, in part, "using the encrypted data file name key *and data file contents* to generate an encrypted data file contents key" (italics added). **Nowhere does Blaze disclose generating a data encryption key from (1) a key used to encrypt a file name and (2) the data contents itself.** Instead, Blaze discloses blocks of data XORed with a bitmask, with the resulting data operated on by Key2.

Even if Blaze's Key1 is used to generate another key, it is not used to encrypt a file name. Even if Key1 is used to generate another key, it is not *file data* used to generate a key.

Contrary to what is stated in the June Office Action, it is not merely the verbiage that distinguishes the claims from Blaze. Specifically:

- Blaze does not disclose using a file name to generate keys;
- Blaze does not disclose generating a data encryption key from a file encryption key AND the data contents itself; and
- A file name and a random number are different things, generated in different ways, and used for different purposes.

C. The Applicants' detailed analysis does not amount to a "general allegation of patentability."

Within the June Office Action, it is stated that, as to Claims 16 and 40, on page 17 of the Response filed electronically on May 18, 2009 (the "May Response"), "[t]he arguments amount to a general allegation of patentability and do not address the rejections applied to the limitations by the Examiner." The Applicants respectfully disagree.

The Examiner ignores the detailed analysis in the rest of the May Response, where the Applicants characterized, in detail, those sections of Blaze cited in the March Office Action. (This analysis is repeated below.) On page 16 of the May Response, the Applicants characterized what Blaze discloses in its Abstract (May Response, page 16, sixth full paragraph) and what Blaze discloses in its section 2.2, fourth paragraph (*id.*, page 16, seventh paragraph). On page 17 of the May Response, the Applicants characterized what Blaze discloses in its section 3, third full paragraph. *Id.*, page 17, first paragraph. Later on page 17, the Applicants summarized what Blaze discloses in all those sections. *Id.*, page 17, second paragraph.

Still on page 17, the Applicants listed specific limitations of Claim 16 and, based on the previous analysis of Blaze, argued that Blaze did not teach these limitations. *Id.*, page 17, fourth paragraph. Only then did the Applicants address the issue of patentability, concluding that the claims were allowable. *Id.*, page 17, fourth paragraph.

This detailed analysis is not, as the Examiner contends, a "general allegation of patentability." By analyzing the cited sections of the prior art in detail, the Applicants did "address the rejections applied to the limitations by the Examiner."

As to Claims 70 and 71, some of the limitations in those claims are included in other claims, such as Claims 1 and 16. Accordingly, the Applicant's characterization of the prior art for Claims 1 and 16 apply to Claims 70 and 71. Again, these characterizations and corresponding analysis are not merely general allegations of patentability. Instead, they also address the rejections raised by the Examiner.

Rejections under 35 U.S.C. § 103(a)

Within the Previous Office Action, Claims 1-5, 9, 11, 12, 19, 20, 26-28, 31, 36, 37, 39, 42, 43, 48-50, and 61-69 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,313,694 to Riedel ("Riedel") in view of Zadok, "Cryptfs: A Stackable Vnode Level Encryption File System" ("Zadok"). The Applicants respectfully disagree.

Claims 1-5, 9, 11, 12, 19, 20, 26-28, 31, 36, 37, 39, 42, 43, 48-50, and 61-69

The independent Claim 1 is directed to a computer system comprising a memory portion containing an encrypted data file and an operating system comprising a kernel. The kernel of Claim 1 comprises a virtual node (a) to *directly* decrypt an encrypted directory entry to determine a location of the encrypted data file and (b) to *directly* decrypt the encrypted data file to access data contained therein. Neither Riedel nor Zadok, either alone or in combination, discloses a virtual node to *directly* decrypt an encrypted directory entry to determine a location of the encrypted data file and to *directly* decrypt the encrypted data file to access data contained therein, as recited in Claim 1. For at least these reasons, the independent Claim 1 is allowable over Riedel, Zadok, and their combination.

Claims 2-5, 9, 11, 12, 19, 20, 61, and 62 all depend on the independent Claim 1. As explained above, the independent Claim 1 is allowable over Riedel, Zadok, and their combination. Accordingly, Claims 2-5, 9, 11, 12, 19, 20, 61, and 62 are all also allowable as depending on an allowable base claim.

The independent Claim 26 is directed to a computer system comprising a first device and a second device. The first device has an operating system kernel and a directory structure with directory information comprising encrypted data file names and corresponding encrypted data file locations for accessing encrypted data files within a file system. The operating system kernel is to decrypt the encrypted data file names and encrypted data file locations using one or more encryption keys to recover clear data corresponding to the data file names, data file locations, and data files. The operating system kernel comprises a virtual node to *directly* encrypt the clear data using the one or more encryption keys to generate cipher data corresponding to the directory information and encrypted data files. The second device is coupled to the first device to exchange cipher data with the first device. Neither Riedel nor Zadok, either alone or in combination, discloses an operating system kernel that comprises a virtual node to *directly* encrypt the clear data using the one or more encryption keys to generate cipher data corresponding to the directory information and encrypted data files, as recited in Claim 26. For at least these reasons, the independent Claim 26 is allowable over Riedel, Zadok, and their combination.

Claims 27, 28, 31, and 63-65 all depend on the independent Claim 26. As explained above, the independent Claim 26 is allowable over Riedel, Zadok, and their combination. Accordingly, Claims 27, 28, 31, and 63-65 are all also allowable as depending on an allowable base claim.

The independent Claim 36 is directed to a method of storing an encrypted data file in a computer file system having a directory. The method of Claim 36 comprises receiving a clear data file having a name and executing kernel code in an operating system, the kernel code comprising a virtual node *comprising drivers* to *directly* encrypt the clear data file to generate an encrypted data file using a symmetric key, store the encrypted data file at a location in the computer file system, and store in the directory an entry containing an encryption of the name and an encryption of the location. Neither Riedel nor Zadok, either alone or in combination, discloses kernel code that comprises a virtual node *comprising drivers* to *directly* encrypt the clear data file to generate an encrypted data file using a symmetric key, store the encrypted data file at a location in the computer file system, and store in the directory an entry containing an

encryption of the name and an encryption of the location, as recited in Claim 36. For at least these reasons, the independent Claim 36 is allowable over Riedel, Zadok, and their combination.

Claims 37, 39, 42, 43, 66, and 67 all depend on the independent Claim 36. As explained above, the independent Claim 36 is allowable over Riedel, Zadok, and their combination. Accordingly, Claims 37, 39, 42, 43, 66, and 67 are all also allowable as depending on an allowable base claim.

The independent Claim 48 is directed to a computer system that comprises a processor, a physical memory containing an encrypted data file and a directory, a secondary device coupled to the physical memory, and an operating system comprising a kernel. The directory comprises a record having a first element corresponding to an encrypted name of the data file and a second element corresponding to an encrypted location of the data file in the memory. The kernel comprises a virtual node integrated with drivers to *directly* decrypt the first and second elements to access the encrypted data file from memory when transferring the data file from the memory to the secondary device and to *directly* re-encrypt the first and second elements when transferring the data file from the secondary device to the memory. Neither Riedel nor Zadok, either alone or in combination, discloses a kernel that comprises a virtual node integrated with drivers to *directly* decrypt the first and second elements to access the encrypted data file from memory when transferring the data file from the memory to the secondary device and to *directly* re-encrypt the first and second elements when transferring the data file from the secondary device to the memory, as recited in Claim 48. For at least these reasons, the independent Claim 48 is allowable over Riedel, Zadok, and their combination.

Claims 49, 50, 68, and 69 all depend on the independent Claim 48. As explained above, the independent Claim 48 is allowable over Riedel, Zadok, and their combination. Accordingly, Claims 49, 50, 68, and 69 are all also allowable as depending on an allowable base claim.

Claims 6-8, 14, 15, 29, 38, 39, 51, and 52

Within the Previous Office Action, Claims 6-8, 14, 15, 29, 38, 51, and 52 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to claim 1, and further in view of U.S. Patent Pub. No. 2003/0005300 to Noble et al. ("Noble"). The Applicants respectfully disagree.

Claims 6-8, 14, and 15 all depend on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claims 6-8, 14, and 15 are all also allowable as depending on an allowable base claim.

Claim 29 depends on the independent Claim 26. As explained above, the independent Claim 26 is allowable. Accordingly, Claim 29 is also allowable as depending on an allowable base claim.

Claim 38 depends on the independent Claim 36. As explained above, the independent Claim 36 is allowable. Accordingly, Claim 38 is also allowable as depending on an allowable base claim.

Claims 51 and 52 both depend on the independent Claim 48. As explained above, the independent Claim 48 is allowable. Accordingly, Claims 51 and 52 are both also allowable as depending on an allowable base claim.

Claims 10 and 30

Within the Previous Office Action, Claims 10 and 30 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok, and further in view of Noble as applied to Claim 5, and further in view of U.S. Patent No. 5,903,881 to Schrader et al. ("Schrader"). The Applicants respectfully disagree.

Claim 10 depends on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claim 10 is also allowable as depending on an allowable base claim.

Claim 30 depends on the independent Claim 26. As explained above, the independent Claim 26 is allowable. Accordingly, Claim 30 is also allowable as depending on an allowable base claim.

Claim 13

Within the Previous Office Action, Claim 13 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to Claim 12, and further in view of U.S. Patent No. 5,727,206 to Fish et al. ("Fish"). The Applicants respectfully disagree.

Claim 13 depends on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claim 13 is also allowable as depending on an allowable base claim.

Claims 16-18, 25, 40, 70, and 71

Within the Previous Office Action, Claims 16-18, 25, 40, 70, and 71 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to Claim 15, and further in view of Blaze, "A Cryptographic File System for Unix" ("Blaze"). The Applicants respectfully disagree.

Riedel and Zadok have been characterized above. Blaze is directed to a Cryptographic File System (CFS). Blaze discloses that "Users associate a cryptographic key with the directories they wish to protect. Files in these directories (as well as their pathname components) are transparently encrypted and decrypted with the specified key without further user intervention; cleartext is never stored on a disk or sent to a remote file server." (Blaze, Abstract) Blaze does not disclose an operating system kernel having a virtual node integrated with drivers to encrypt and decrypt data.

In section 2.2, fourth paragraph, cited in the Office Action, Blaze discloses protecting a directory using a set of cryptographic keys, passphrases entered from a keyboard. Blaze discloses, generally, that the passphrases are used to generate several independent keys.

In section 3, third paragraph, cited in the Office Action, Blaze discloses using a passphrase to generate 2 keys. Specifically, Blaze discloses using the first key to compute a bit mask for masking a part of a file block. The result is then encrypted with the second key. The result is not used to generate the second key.

Specifically, Blaze does not disclose taking a pass key *and a data file name* to generate an encrypted data file name key. Blaze also does not disclose taking that encrypted file name data key *and data file contents* to generate another key, an encrypted data file contents key.

Claims 16-18 and 25 all depend on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claims 16-18 and 25 are also allowable as depending on an allowable base claim.

Claim 16 is allowable for at least one additional reason. Claim 16 recites, "a key engine to receive a pass key and a data file name to generate an encrypted data file name key, the key engine also to use the encrypted data file name key and data file contents to generate an encrypted data file contents key, the key engine also to encrypt the data file contents with an encrypting data file contents key to generate encrypted data file contents and to encrypt the data file name with an encrypting data file name key to generate an encrypted data file name" (italics added). Not one of Riedel, Zadok, Blaze, and their combination teaches this element. For this additional reason, Claim 16 is allowable over Riedel, Zadok, Blaze, and their combination.

Claim 16 has been amended to recite that the previously generated encrypted data file contents key and encrypted data file name key are used to do perform later encryptions. In a previous response, the Applicants explained how a key that is encrypted can also be used to perform later encryptions. (*See*, Response filed electronically on December 22, 2008, at page 11, 3rd full paragraph). This amendment does not add new matter, finding support in the original Claim 16.

Claim 40 depends on the independent Claim 36. As explained above, the independent Claim 36 is allowable. Accordingly, Claim 40 is also allowable as depending on an allowable base claim.

Claim 40 is also allowable for an additional reason. Claim 40 recites, "wherein executing kernel code comprises entering a pass key and a data file name into a first encryption process to produce an encrypted data file name and an encrypted data file name key; and processing the file contents together with the encrypted data file name key to generate an encrypted file contents key and an encrypted file contents." Not one of Riedel, Zadok, and Blaze, either alone or in combination discloses this element. For this additional reason, Claim 40 is allowable.

The independent Claim 70 is directed to a computer system containing an operating system. The computer system of Claim 70 comprises a kernel, a memory, and an encryption key management system. The kernel comprises a virtual node integrated with drivers to encrypt and decrypt data transferred between a memory and a secondary device. The kernel also comprises an encryption engine to encrypt clear data to generate cipher data. The encryption engine is also to decrypt the cipher data to generate the clear data. The memory is coupled to the encryption engine to store the cipher data and comprises a first logical protected memory to store encrypted file data and a second logical protected memory to store encrypted key data. The encryption key management system is to control access to the encrypted file data and the encrypted key data. The encryption key management system comprises a key engine to receive a pass key and the file name to generate an encrypted file name key, use the encrypted file name key and file contents to generate an encrypted file contents key, and encrypt the file contents with an encrypting file contents key to generate encrypted file contents. Not one of Riedel, Zadok, Blaze, and their combination discloses a key engine to receive a pass key *and the file name* to generate an encrypted file name key, use the encrypted file name key *and file contents* to generate an encrypted file contents key, and encrypt the file contents with an encrypting file contents key to generate encrypted file contents. For at least these reasons, the independent Claim 70 is allowable over Riedel, Zadok, Blaze, and their combination

The independent Claim 71 is directed to a method of encrypting data. The method of Claim 71 comprises receiving clear data and executing kernel code in an operating system. The kernel code comprises a virtual node integrated with drivers to use a symmetric key to encrypt the clear data to generate cipher data and to use the symmetric key to decrypt the cipher data to generate the clear data. The executing the kernel code comprises entering a pass key and a file name into a first encryption process to produce an encrypted file name and an encrypted file name key and processing the file contents together with the encrypted file name key to generate an encrypted file contents key and encrypted file contents. Not one of Riedel, Zadok, Blaze, and their combination discloses that executing kernel code comprises entering a pass key *and a file name* into a first encryption process to produce an encrypted file name and an encrypted file name key. Not one of Riedel, Zadok, Blaze, and their combination discloses processing *the file contents* together with the encrypted file name key *to generate an encrypted file contents key and encrypted file contents*. For at least these reasons, the independent Claim 71 is allowable over Riedel, Zadok, Blaze, and their combination.

Claims 21, 32, and 44

Within the previous Office Action, Claims 21, 32, and 44 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to Claim 19, and further in view of U.S. Patent No. 6,836,888 to Basu et al. ("Basu"). The Applicants respectfully disagree.

Claim 21 depends on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claim 21 is also allowable as depending on an allowable base claim.

Claim 32 depends on the independent Claim 26. As explained above, the independent Claim 26 is allowable. Accordingly, Claim 32 is also allowable as depending on an allowable base claim.

Claim 44 depends on the independent Claim 36. As explained above, the independent Claim 36 is allowable. Accordingly, Claim 44 is also allowable as depending on an allowable base claim.

Claims 22-24, 33-35, 45, and 47

Within the Previous Office Action, Claims 22-24, 33-35, 45, and 47 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to Claim

19, and further in view of U.S. Patent No. 6,477,545 to LaRue ("LaRue"). The Applicants respectfully disagree.

Claims 22-24 all depend on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claims 22-24 are all also allowable as depending on an allowable base claim.

Claims 33-35, 45, and 47 all depend on the independent Claim 26. As explained above, the independent Claim 26 is allowable. Accordingly, Claims 33-35, 45, and 47 are all also allowable as depending on an allowable base claim.

Claim 41

Within the Previous Office Action, Claim 41 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to Claim 40, and further in view of Noble. The Applicants respectfully disagree.

Claim 41 depends on the independent Claim 36. As explained above, the independent Claim 36 is allowable. Accordingly, Claim 41 is also allowable as depending on an allowable base claim.

Claims 59 and 60

Within the Previous Office Action, Claims 59 and 60 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Riedel in view of Zadok as applied to Claim 1, and further in view of U.S. Patent No. 6,938,166 to Sarfati et al. ("Sarfati"). The Applicants respectfully disagree.

Claims 59 and 60 both depend on the independent Claim 1. As explained above, the independent Claim 1 is allowable. Accordingly, Claims 59 and 60 are both also allowable as depending on an allowable base claim.

The new Claim 72 is allowable.

The new Claim 72 depends on the independent Claim 1. As explained above, Claim 1 is allowable. Accordingly, Claim 72 is also allowable as depending on an allowable base claim.

The new Claim 72 recites, in part, "a plurality of different encryption keys to decrypt corresponding blocks of the data file." This limitation finds support in the application, such as at page 41, line 23, to page 42, line 4, which explains that data can be broken into a series of blocks, each encrypted with its own encryption key.

The new Claim 73 is allowable.

The new independent Claim 73 is directed to a computer system comprising a memory portion containing an encrypted data file and an operating system comprising a kernel. The kernel comprises a virtual node to decrypt an encrypted directory entry to determine a location of the encrypted data file and to decrypt the encrypted data file to access data contained therein. The virtual node is to decrypt the data file using a first key generated from an identifier of the operating system, an identifier of a file system containing the data file, an identifier of a root directory containing the encrypted data file, and identifier of the data file.

As explained above, none of the cited prior art, either alone or in combination, discloses a kernel that comprises a virtual node, such that the virtual node decrypts an encrypted directory entry to determine a location of the encrypted data file and to *directly* decrypt the encrypted data file to access data contained therein. Furthermore, none of the cited prior art, either alone or in combination, discloses a virtual node to decrypt the data file using a first key generated from an identifier of the operating system, an identifier of a file system containing the data file, an identifier of a root directory containing the encrypted data file, and identifier of the data file. For at least these reasons, the new Claim 73 is allowable.

The new Claim 73 does not add new matter. It contains the limitations of the previously pending Claim 1, as well as limitations that find support in the application, such as at page 64, lines 1-8.

CONCLUSION

For the reasons given above, the Applicants respectfully submit that Claims 1-45, 47-52 and 59-73 are in condition for allowance, and allowance at an early date would be appreciated. If the Examiner has any questions or comments, the Examiner is encouraged to call the undersigned at (408) 530-9700 so that any outstanding issues can be quickly and efficiently resolved.

Respectfully submitted,
HAVERSTOCK & OWENS LLP

Dated: July 6, 2009

By: /Jonathan O. Owens/
Jonathan O. Owens
Reg. No.: 37,902
Attorneys for Applicants